

Пояснительная записка

Описание

Номер задания: 4

Номер функции: 21

Реализовать контейнер объемных (трехмерных) геометрических фигур с вариациями: шар, параллелепипед, правильный тетраэдр.

Входные данные – тип фигуры и плотность ее материала, обозначаемые целым и вещественным числом соответственно.

Типы фигур:

1. Шар: целочисленный радиус
2. Параллелепипед: три целочисленных ребра
3. Правильный тетраэдр: целочисленная длина ребра

Вычислить площадь поверхности каждой фигуры и удалить из контейнера те элементы, у которых площадь поверхности меньше, чем среднее арифметическое площадей поверхности для всех элементов контейнера. Остальные элементы передвинуть в начало контейнера с сохранением порядка.

Метрики

Программа состоит из 6 модулей реализации (.cpp) и 6 интерфейсных модулей (.h).

Полученный размер исполняемого кода: 15.0 Кб.

Размер исполняемого файла: 63.0 Кб.

Время работы случайной генерации фигур:

-10 фигур: 2 мс

-100 фигур: 4 мс

-1000 фигур: 12 мс

-10000 фигур: 75 мс

Сравнение с написанными ранее программами

Программа, написанная с использованием ООП подхода, выполняется быстрее, чем программа с использованием процедурного подхода. Пример выполнения случайной генерации 10000 фигур в ООП и процедурном подходах:

ООП – 75 мс

Процедурный – 219 мс

Таким образом, время работы уменьшилось почти в 3 раза.

Структурная схема

ТАБЛИЦА ТИПОВ

Int – 4 Double – 8
class Parallelepiped 12 a – int 4[0] b – int 4[4] c – int 4[8]
class Sphere 4 r – int 4[0]
class Tetrahedron 4 a – int 4[0]
class Random 8 first – int 4[0] last – int 4[4]
class Shape 32 rnd20 – Random 8[0] rnd3 – Random 8[8] rnd100000 – Random 8[16] materialDencity – double 8[24]
class Shape 32 rnd20 – Random 8[0] rnd3 – Random 8[8] rnd100000 – Random 8[16] materialDencity – double 8[24]

ПАМЯТЬ ПРОГРАММЫ

Shape* Shape::StaticIn(FILE* file, int k) 20 file – FILE* 8[0] k – int 4[8] sp – Shape* 8[12] materialDencity – double
Shape* Shape::StaticInRnd() 20 K – int 4[0] materialDencity – double 8[4] sp – Shape* 8[12]
void Container::In(FILE* file) 8 file – FILE* 8[0] k – int storage – Shape* len – int
void Container::InRnd(int size) 4 size – int 4[0] len – int storage – Shape*
void Container::Out(FILE* file) 8 file – FILE* 8[0] storage – Shape* len – int

Схема работы программы

